Amendments to the Claims

1(Currently amended). A method of implementing ~~the~~ a Viterbi algorithm comprising:

calculating branch metrics for branches of ~~the~~ a Viterbi trellis;

combining the branch metrics with old path metrics to produce candidate path metrics;

selecting a new path metric associated with a selected trellis path for each state in the Viterbi trellis from the candidate path metrics; and

composing an optimal path value for each state in the Viterbi trellis, the optimal path value indicating multiple transitions of the selected trellis path.

2(Currently amended). The method of Claim 1 wherein the optimal path value ~~is comprised of~~ comprises optimal path value fragments.

3(Original). The method of Claim 2 wherein the optimal path value fragments are stored in memory blocks.

4(Original). The method of Claim 1 further comprising using the optimal path value to determine an output value.

5(Currently amended). The method of Claim ~~4~~ 1 wherein the optimal path value for ~~the~~ a lowest state of the Viterbi trellis at ~~the~~ an end of a memory block ~~of symbols~~ indicates ~~the~~ transmitted symbols ~~in the block~~.

6(Canceled).

7(Canceled).

8(Canceled).

9(Currently amended). The method of Claim 1 ~~wherein~~ further including using a traceback pointer ~~is used~~ to select between two prior optimal path values.

10(Currently amended). The method of Claim 9 ~~wherein the~~ further including selecting an old optimal path ~~values are selected~~ from the optimal path ~~values~~ value for ~~states~~ a state that ~~can transition~~ transitions into a current state.

11(Canceled).

12(Currently amended). The method of Claim 1 wherein the Viterbi algorithm is implemented ~~in~~ by a reconfigurable chip.

13(Currently amended). The method of Claim 1 wherein selecting the new path metric ~~selecting~~ and composing the optimal path value ~~composing steps~~ operate in parallel.

14(Currently amended). An apparatus to implement ~~the~~ a Viterbi algorithm comprising:

a path metric storage ~~adapted~~ to store a path metric associated with a selected trellis path for each state in ~~the~~ a Viterbi trellis:

a path update unit ~~adapted~~ to update each path metric;

an optimal path value storage ~~adapted~~ to store an optimal path value for each state in the Viterbi trellis, the optimal path value indicating multiple transitions of the selected trellis path; and

~~a~~ an optimal path value update unit ~~adapted~~ to update each optimal path value.

15(Currently amended). The apparatus of Claim 14 further comprising an output unit ~~adapted to use an optimal path value~~ to determine an output value for the Viterbi algorithm.

16(Currently amended). The apparatus of Claim 15 wherein the output value ~~produces~~ is the optimal path value for the state with ~~the~~ a lowest path metric.

17(Canceled).

18(Currently amended). The apparatus of Claim 14 wherein the path ~~metric~~ update unit and the optimal path value update ~~operates~~ unit operate in parallel.

19(Currently amended). The apparatus of Claim 14 ~~implemented on~~ wherein the Viterbi algorithm is computed using reconfigurable logic.

20(Currently amended). The apparatus of Claim 19 wherein the optimal path ~~updating~~ value update unit uses resources not ~~required for~~ used by the path ~~metric~~ update unit.

21(Currently amended). The apparatus of Claim 14 wherein there is no serial traceback operation ~~is required~~.

22(Currently amended). The apparatus of Claim 14 wherein the optimal path value storage ~~is separated into~~ includes multiple memory blocks.

23(Currently amended). The apparatus of Claim 22 wherein the optimal path value comprises optimal path fragments stored in the multiple memory blocks.

24(Currently amended). The apparatus of Claim 23 wherein the optimal path value ~~updating unit operates on~~ update unit operates on the optimal path value fragments to update the optimal path value.

25(Original). The apparatus of Claim 23 wherein, during the readback operation, the optimal path fragment from one memory block is used to determine a pointer to an optimal path fragment in another memory block.

26(New). A method to determine a maximum-likelihood path through a convolutional encoder, comprising:

a parallel implementation to prefetch previous optimal paths of two states from an optimal path memory and fetch traceback path width data bits from a path memory that is appended with data bits to provide an entire optimal path leading up to that path for each state.

27(New). The method of claim 26 further comprising:

sharing address generators for the optimal path memory and the path memory.

28(New). The method of claim 26, further comprising:

receiving a new symbol by the convolutional encoder; and

appending a new bit to the optimal path stored in the optimal path memory for each new symbol received.